Applicant(s): NATAN VISHLITZKY, DOUGLAS E. LECRONE, IZHAR SHARON, DANIEL A. MURPHY, WILLIAM R. FAIRCHILD, HANA MORESHET, MARTIN FARLEY AND ELIZABETH C. PATAPOUTIAN
Serial No.: Continuation of 09/731,245
Filed: Herewith

In the Specification

Insert the following paragraph at page 1, line 1:

Cross Reference to Related Application

This application is a continuation of co-pending United States Patent Application Serial No. 09/731,245, filed December 6, 2000 entitled Method for Enabling Overlapped Input/Output Requests to a Logical Device Using Assigned and Parallel Access Unit Control Blocks.

Please replace the paragraph beginning at page 8, line 1 with the following paragraph:

In accordance with this invention a host can generate overlapped input-output requests for a logical volume in a disk array storage device. The host operating system normally utilizes a first, uniquely identified, base unit control block corresponding to the logical volume to effect a transfer in response to the input-output request. Within the host there are defined at least one related, uniquely identified unit control block that identifies the logical volume. A parallel access control block is established for each unit control block associated with the logical volume and a parallel access main

control block for the logical volume through which each of the base and related unit control blocks can be identified. The host responds to the input-output request by interrupting the operating system and assigning one of the base and related unit control blocks to the input-output request. Then the host returns control of the response to the input-output request to the operating system identifying the assigned unit control block whereby the host operating system can issue overlapped input-output requests to the given logical volume. Within the disk array storage facility a table is established for the logical volume with an entry input-output requests and corresponding parameters. Parameters for each new input-output request to the logical volume are tested with respect to the parameters for input-output request entries in the table. The test determines which, of a plurality of control functions including enabling the processing of the input-output request by the storage facility, will be performed.

Please replace the paragraph beginning at page 18, line 5 with the following paragraph:

All the foregoing procedures are conventional MVS operating procedures that are well known in the art. In

accordance with this invention, an operating system, such as an MVS operating system can be adapted to provide the advantages of parallel access by adding certain features of this invention to the conventional operating system and by modifying the process by which the primary data storage system 33 handles commands received from a host control processor. Further it has been found that these modifications enable three additional features to be realized. It is possible to reduce the size of a defined extent to a required extent that represents the actual extent of tracks that I/O requests in a command chain will use. It is also possible to eliminate write serialization from [____] I/O requests that are actually read-only. It is further possible to accommodate requests from different host processors to a single logical device. These four features, individually and in different combinations, can improve the rate at which data transfers occur between the host processors and a logical device.

Please replace the paragraph beginning at page 24, line 7 with the following paragraph:

Step 86 creates the control blocks 85 including a PAVCVT control block 87 shown in FIG. [5 ]2. This is a primary control block from which any other control block in the PAV

subsystem 83 can be reached. More specifically, step 86 in FIG. 4 creates the PAVCVT control block 87 with a structure as shown in FIG. 5. It also creates PAVB and PAVA control blocks having structures as shown in FIGS. 6 and 7. For the specific configuration file listed above, step 86 creates, for the first logical device, one PAVB control block 88 for the base device C06C and two PAVA control blocks 89 and 90. It also creates one PAVB and two PAVA control blocks for the base device C06D and one PAVB and two PAVA control blocks for base device C06E. These have the same structure. They are imbedded in the control blocks 85, but are not shown.

Please replace the paragraph beginning at page 25, line 13 with the following paragraph:

Step 97 in FIG. 4 loads the EMC_STARTIO module 65 into the common address space 50. As previously stated, the EMC_STARTIO module 65 operates before the MVS_STARTIO module 56. Step 97 also loads the ptr_MVS_STARTIO pointer 55 and the ptr_EMC_STARTIO pointer 66 into locations 98 and 99 in FIG. 5, respectively. Step 100 in FIG. 4 loads an EMC_IOSVSCP module 67 into the common address space and the ptr_MVS_IOSVSCP pointer [67] 57 and the ptr_EMC_IOSVSCP pointer 68 into locations 101 and 102, respectively. Similarly, step 103 loads

the EMC_I/O_INTERRUPT_TRACE module 69 into the common space 50 of FIG. 2 to be used before the MVS_I/O_INTERRUPT_TRACE module 64. Step 103 additionally loads pointers to the MVS_I/O_INTERRUPT_TRACE module 64 and an EMC_I/O_INTERRUPT_TRACE module 69 into locations 104 and 105, respectively. This completes the process by which the PAV application is readied to respond to I/O requests in accordance with this invention.

Please replace the paragraph beginning at page 28, line 7 with the following paragraph:

Location 119 contains the address of a first PAVA control block location. In this specific example, location 119 contains a pointer to a PAVA control block associated with the alias UCB for an unused device C078. Location 120 identifies the number of alias *UCB*'s associated with the base UCB. In the specific example the PAVB control block for the base UCB C06C contains a "2". Location 121 contains a volume serial number as known in the art Location 122 contains an address that points to the next one of the alias PAVA control blocks for requeueing as described more fully later.

Applicant(s): NATAN VISHLITZKY, DOUGLAS E. LECRONE, IZHAR SHARON, DANIEL A. MURPHY, WILLIAM R. FAIRCHILD, HANA MORESHET, MARTIN FARLEY AND ELIZABETH C. PATAPOUTIAN
Serial No.: Continuation of 09/731,245
Filed: Herewith

Please replace the paragraph beginning at page 38, line 17 with the following paragraph:

If neither of these conditions exists, step 202 and step 203 transfer control to step 204 that initializes MAXFOUND and MINFOUND registers 205 and 206 in a PSQWK work space 207 shown in FIG. 11 for use by the EMC_IOSVSCP module 67. In one particular embodiment step 204 initializes the MAXFOUND register 205 to a low value, such as X'00' and the MINFOUND register 206 to a high value such as X'FF'. Step [210] 208 completes the initialization by clearing a DEFINE EXTENT DATA CHANGED flag 116 in a corresponding one of the PAVB or PAVA control blocks of FIGS. 6 and 7 and a WRITE COMMAND FOUND flag 209 in FIG. 11.

Please replace the paragraph beginning at page 44, line 10 with the following paragraph:

When this feature is combined with the parallel access feature of FIGS. 4 through 9, significant improvements in access can be achieved. For example, assume a file is allocated to all the tracks in cylinders 50 through 99 and that there are multiple jobs attempting to access this file simultaneously, some reading and some writing into it. Assume also that the Define Extent command specifies all fifty of

these cylinders. I/O requests typically transfer only one block at a time from one track within one cylinder. If an I/O request only needs to write data into cylinder 55, track 8 and the Define Extent command covers all fifty cylinders, then an I/O request that wants to read data from cylinder 97 will have to wait until the first I/O request completes. If, on the other hand, the processes in FIGS. 10A through 10C determine that collectively all the channel command words in a particular I/O request are limited to accessing data from cylinder 55, track 8, then the read operation from cylinder 97 will not have to wait until the write I/O operation completes. Thus each chain of channel commands that is transmitted to the primary data storage system 33 in FIG. 1 will include a shrunk extent in accordance with values established by the actual data to be transferred and with the [_____] intent parameter set in accordance with the actual commands in the I/O request. The EMC_IOSVSCP module 67 of FIGS. 10A through 10C terminates with step 230. When this occurs, control passes to the MVS_IOSVSCP module 58 in FIG. 2 to initiate the I/O request using the altered address extent and other parameters if optimization has occurred.

Applicant(s): NATAN VISHLITZKY, DOUGLAS E. LECRONE, IZHAR SHARON, DANIEL A. MURPHY, WILLIAM R. FAIRCHILD, HANA MORESHET, MARTIN FARLEY AND ELIZABETH C. PATAPOUTIAN
Serial No.: Continuation of 09/731,245
Filed: Herewith

Please replace the paragraph beginning at page 59, line 11 with the following paragraph:

FIG. 17 depicts the EXTENT_IS_OVERLAPPED module 312 that begins by setting an initial return value of "0" in step 379. Step 380 tests the SYNC flag 331 in the selected entry from the extent queue table 301. If that flag is set, the return is set to an EX_Q_FORCE_OVERRUN value in step 381, and the module 312 terminates its operation. This module can be called at other times within the processing of one of the predetermined commands. At this particular time, however, the SYNC flag will not be set. If it had been, prior analysis would have prevented the process from proceeding to this point.

Please replace the paragraph beginning at page 60, line 1 with the following paragraph:

When the SYNC flag is not set, the module tests the SYNC flag in the new entry that is being analyzed in step 382. If that SYNC flag is set, the return is set to an EXT_Q_OVERLAP value in step 383. Again, as any entry with the SYNC flag set must be handled to the exclusion of all other entries, no additional analysis is needed.